15-451/651 Algorithms, Fall 2017

Homework #5 **Due: November 5, 2017**

This HW has three regular problems, and one programming problem. All problems on written HWs are to be done *individually*, no collaboration is allowed.

Solutions to the three written problems should be submitted as a single PDF file using `gradescope`, with the answer to each problem starting on a new page.

Submission instructions for the programming problem will be posted on the website and Piazza.

(25 pts) 1. **Shortest Cycle**

You're given a directed weighted graph $G = (V, E, w)$. There are no self-loops or multi-edges. Here $w_{i,j} \geq 0$ is the length of edge $(i, j) \in E$. For $(i, j) \notin E$ we have $w_{i,j} = \infty$.

Your goal is to find the length of the shortest cycle in G.

(a) Give an algorithm that runs in $O(nD(n, m)$, where $D(n, m)$ is the running time of Dijkstra's algorithm on a graph of n vertices and m edges.

(b) Another solution to this problem is obtained by tweeking the Floyd-Warshall algorithm. Initialize $A^0_{i,j} = w_{i,j}$. Note that the diagonal elements are all initially ∞ (unlike the standard version of Floyd-Warshall).

For $k = 1, 2, \ldots n$ do:
    For $i = 1, 2, \ldots n$ do:
        For $j = 1, 2, \ldots n$ do:
            $A^k_{i,j} \leftarrow \min(A^{k-1}_{i,j}, A^{k-1}_{i,k} + A^{k-1}_{k,j})$

Prove that after running this $O(n^3)$ algorithm the minimum length cycle is $\min_{1 \leq i \leq n} A^n_{i,i}$.

(c) (bonus) Can you give an algorithm that is more efficient than both of these methods?

(25 pts) 2. **Triangle Game**

Let $T = (a, b, c)$ be a triangle in the plane where $a = (-1, 0)$, $b = (1, 0)$ and $c = (0, 2)$. Player I (the "evader") and Player II (the "searcher") play the following zero-sum game. The evader picks a point X in T and simultaneously the searcher picks a point Y in T. The payoff to the evader is $\|X - Y\|^2$, the square of the Euclidean distance between X and Y.

Find the value v of this game (from the evader's viewpoint). Prove that it's correct by giving a strategy for the evader that achieves at least v in expectation, and one for the searcher that prevents the evader from earning more than v in expectation.

(25 pts) 3. **Ice Cream**

The great nation of Glacia has hired you as a contractor to figure out how to maximize the national production of ice cream, the country's primary export. The country has a set of cities S of size $|S| = n$, where each city has either some cream-producing farms, or some factories to produce and package the ice cream, or neither. In particular, city i can produce r_i units of cream and can process f_i units of ice cream in its factories, where either $r_i = 0$ or $f_i = 0$ (or both) for every city.

However, due to local agreements with various shipping companies, it isn't always possible to move resources from every city to every other city. In particular, there are m shipping companies, and shipping company j has agreements with a set of cities $c_j \subseteq S$ and may ship freely into and out of any city in this set, but not to any other cities. So that the cream doesn't go bad, at most two shipping companies may handle a given unit of cream to transport it from its source farm to its destination city. The goal is to decide which shipping companies should move cream between which cities to maximize the national production of ice cream.

For example, suppose there are four cities, 1, 2, 3, 4, and 5, with the following cream-production and cream-processing capacities:

City	r_i	f_i
1	6	0
2	2	0
3	0	0
4	0	5
5	0	3

There are three shipping companies, A, B, and C, with the following sets of agreements:

Shipping Company	c_j
A	$\{1, 3, 4\}$
B	$\{2, 3, 4\}$
C	$\{2, 5\}$

In this case, an optimal arrangement provides a total production of 7 units of ice cream, using A to ship 5 units of cream from 1 to 4 and using C to ship 2 units of cream from 2 to 5. Note that we're not allowed to have A ship 1 unit of cream from 1 to 3 that B transports to 2 and C transports to 5; however, if C had an agreement with city 3, then we could augment the above arrangement by having A transport 1 unit from 1 to 3 which C then transports to 5 a total production of 8.

This problem lends itself well to a network flow solution; however, it turns out that there are actually two good constructions, and which is better is dependent on the magnitude of the number of cities n as compared to the number of shipping companies m!

(a) Provide a network flow construction (that is, describe how to construct the network graph) that works well when $m \gg n$ (e.g., $m > n^2$). In terms of big-O, how many nodes and edges does this construction have?

(b) Provide a *different* network flow construction that works well when $m \ll n$ (e.g., $n > m^2$). In terms of big-O, how many nodes and edges does this construction have?

Hint. When $n = m$, the two constructions have an asymptotically-equal number of nodes and edges.

(25 pts) 4. **CATS**

In this programming problem you're given a rectangular board of letters. The word "cats" appears in the board if there is a sequence of four squares such that "c" is in the first one, "a" is in the second one, and so on. Each letter is next to (either horizontally or vertically) the previous one.

The *score* of a board is the maximum number of times you can find "cats" in it. That is, you find the word "cats" as described above. Then cross off those cells (they can't be used again). Then repeat this step as many times as possible. (See the examples below.) The time limit is 10 seconds.

Input:

The first line contains two space-separated integers: r and c. The next r lines are strings of c lower case letters followed by a newline. $1 \leq r, 1 \leq c$, and $r \cdot c \leq 10000$.

Output:

Print the maximum number of disjoint cats that can be simultaneously found on the board (the score described above).

Here are some sample input and output pairs:

`4 4`
`cats`
`axxx`
`txxx`
`sxxx`
	`1 cat`

`1 10` `stacatstac`	`2 cats`

`4 4`
`saos`
`tacc`
`tcoa`
`otcs`
	`1 cat`

`2 10`
`catscatsca`
`atscatscat`
	`3 cats`

`1 1` `x`	`0 cats`